Computer Science Department

TECHNICAL REPORT

# THE LIMITS OF ARTIFICIAL INTELLIGENCE
Article for 'Encyclopedia of Artificial Intelligence'

By Jacob T. Schwartz

Technical Report #212
March 1986

NEW YORK UNIVERSITY

Department of Computer Science
Courant Institute of Mathematical Sciences
251 MERCER STREET, NEW YORK, N.Y. 10012

# THE LIMITS OF ARTIFICIAL INTELLIGENCE

Article for 'Encyclopedia of Artificial Intelligence'

By Jacob T. Schwartz

Technical Report #212
March 1986

# The Limits of Artificial Intelligence

Article for 'Encyclopedia of Artificial Intelligence'

J.T. Schwartz
Robotics Activity
Courant Institute
New York University

## 1. Introduction

The question of what intrinsic limits constrain the artificial intelligence enterprise, which can be defined as the attempt to construct electronic systems exhibiting human or superhuman levels of capability in areas traditionally regarded as mental, has been debated within very wide limits. On one side one finds a substantial community of researchers who believe firmly that such systems will prove possible. Their common (but not universal) assumption is that the organic brain is in effect a complex electrochemical system operating in some (doubtless highly parallel) but essentially computer-like fashion, and hence gives direct proof of the realizability of intelligence by mechanism; *vide* Marvin Minsky's flat-footed 'The brain is a meat machine'. Opposing this view one finds the assertion that mental processes are essentially indecomposable, lie outside the narrow reach of scientific reductionism, and that their indecomposability sets fundamental limits to any attempt to duplicate intelligence by mechanism. From this point of view, e.g. as represented by the writings of Hubert Dreyfus, the history of artificial intelligence research to date, consisting always of very limited success in particular areas, followed immediately by failure to reach the broader goals at which these initial successes seem at first to hint, gives empirical proof of the presence of irreducible wholes fundamentally incapable of being comprehended, much less duplicated, by the narrowly technical procedures of artificial intelligence researchers.

This philosophical debate concerns the existence of *fundamental* limits to the artificial intelligence enterprise, which however is only one of several kinds of potentially significant limit that need to be considered. Even if no such fundamental limits existed, i.e. even if a hypothetical infinitely fast computing engine possessed of infinite amounts of memory could in principle duplicate all aspects of human mental capability, it would still remain necessary to ask just how much computation and data storage such duplication would require. Suppose, for example, that it could be shown that the minimum computational resource required to duplicate some human mental function is implausibly large, relative either to the extreme limits of physically realizable computation, or to the largest computers likely to be constructed over the next decades or centuries. In this case, construction of significant artificial intelligences would be blocked by inescapable practical limits, even if fundamental limits did not exist. Finally, even if no such *computational* factors proved to limit the possibility of artificial intelligence, one would still want to assess the existing state of the field and project the rate of progress likely to result from application of its present intellectual tools to the profound problems with which it must wrestle.

The next five sections of the present article develop points relevant to the three kinds of limits defined in the preceding paragraph. A final section discusses certain other concerns, implicit in the debate between the enthusiasts of artificial intelligence and their opponents, which may explain some of the vehemence which has crept into this debate.

## 2. The Question of Fundamental Limits to the Constructability of Artificial Intelligences

### 2.1. A Very Brief Comment on the Philosophical Issue

In his deservedly famous 1950 article, Alan Turing proposed to replace amorphous philosophical debate about whether machines could 'really' think by the more pragmatic question of whether they could imitate the behavior of thinking beings well enough to make the assumption that they are 'thinking' the most comfortable basis for continuing interaction with them. The practical force of Turing's argument seems overwhelming. If at some future time people find themselves surrounded by artificially produced beings capable of performing the same variety of daily tasks, physical and intellectual, that one would expect of a person, and in particular capable of conversing on an unrestricted variety of topics in entirely easy, flexible manner, artificial intelligence will have been attained. This is not to deny the possibility that humans in this situation may choose to regard themselves as a kind of nobility, distinguished in view of their long and imperfectly understood biological pedigree from more fully understood and easily repairable/replaceable creatures. Such an attitude can even find objective justification in the reflection that, as long as any significant aspects of human function remain incompletely understood, humanity incorporates a pool of capabilities, tested by long evolution, which deserves protection and cautious nurture proportional to its long history and mysterious potential; these strong points also apply to whales and snail-darters.

Nevertheless, in the real presence of robots exhibiting human levels of flexibility and capability, the question as to whether these beings 'really' thought or merely 'appeared to' think and feel would lose pragmatic force, though of course its ideological importance might grow, perhaps even greatly. It makes less sense for the present article to pursue this debate than to assess the probability that such a situation will really arise.

## 2.2. The Brain as a Biochemical Computer

As already noted, part of the confidence with which artificial intelligence researchers view the prospects of their field stems from the materialist assumption that 'mind' is simply a name for the information-processing activity of the brain, and that the brain is a physical entity which acts according to the laws of biochemistry in a manner uninfluenced by any irreducible 'soul' or other unitary, purely mental entity incapable of analysis into a causal sequence of elementary biochemical events. Compelling evidence for the equation of mental function with the physical activity of the brain is easily drawn from many branches of science, and in particular from experimental neurobiology. For example, discrete lesions at the rear of the cerebral cortex produce discrete blind spots (scotomas) in the visual field, which turns out to communicate in 1-1 continuous fashion with the family of sensory neurons comprising the retina of the eye. Similarly, stimulation of points on the upper central portions of the cortex (temporal motor area) will produce elementary twitching motions of particular muscles. Physical manipulation of nervous tissue can also generate and/or remove sensations having profound motivational significance, e.g. direct application of an excess of potassium to the cutaneous nerves causes sharp pain; conversely, application of Novocaine to an appropriate branch of the facial nerve blocks dental pain in particular areas, thus permitting dental manipulations which would be unbearably aversive were the nerves communicating this sensation of pain not 'turned off'. These elementary remarks, plus thousands of far more precise observations obtained by direct recording of the electrical activity of individual neurons, show that neuronal activity reflects external stimuli and behavior (even intended behavior before its overt expression) in detailed and quantitative fashion, at least for those sensory and motor systems for which such correlations can be expected *a priori* to be understood most easily.

As might also be expected, detailed understanding of the manner in which neuronal activity reflects and governs a living creature's interactions with its environment is most complete for the simplest animals, particularly those whose nervous systems consist of relatively few neurons which, being particularly large, are relatively easy to identify and examine individually. A typical but particularly well-studied example of this is the marine snail *Aplysia Californica*, whose nervous system consists of roughly 20,000 neurons divided into nine separate ganglia within which hundreds of individual cells have been specifically identified. Fairly detailed understanding of the patterns of neuronal activity and interconnection governing many of the most typical and vital reactions of this simple creature has been attained. For example, much is known about the manner in which its nervous system controls heartbeat, respiration, gill withdrawal reflex, release of ink in response to a sensed danger, feeding, reproduction, etc. Moreover, *Aplysia* is capable of certain rudimentary types of learning (including *sensitization*, which progressively increases reactions to certain stimuli, and *habituation*, which progressively reduces other reactions), and the biochemical bases for these forms of neuronal plasticity have been at least partly elucidated. Finally, the nervous activities controlling sensation and behavior in *Aplysia* have been shown to be inherent properties of the nervous system which persist even when this system is dissected out of the body of *Aplysia* and maintained artificially in a suitable nutrient bath, provided that the afferent signals expected along certain sensory nerves are supplied electrically after the sensory organs that would normally give rise to them have been removed. In this last case, the analogy with a robot's computer brain detached from its body and running in an artificially emulated environment of input and proprioceptive signals is overwhelming.

Some may object to facile extrapolation from the reflexive behaviors of a simple 20,000-neuron creature to the vastly more sophisticated activity of the roughly hundred billion neurons of the human brain. Nevertheless, the (admittedly highly incomplete) biological evidence available thus far seems to favor just such an extrapolation: living creatures of whatever complexity seem to share a common neuronal biochemistry in much the same way that they share a common genetic code.

Thus what neurobiological evidence there is hints strongly that no difference of fundamental principle separates the brain from any other other form of computer, or can be expected to limit the range of possibilities which artificial intelligence research can legitimately explore.

### 2.3. Quantitative Estimates Concerning the Brain's Computing Power

Even the resolutely mechanistic conclusion drawn in the preceding subsection leaves open two possibilities, either of which could still rule out the possibility of attaining human-like levels of mental capability by artificial means. In the first place, the mass of computational activity performed each second by the living brain, and/or the mass of information available to the brain for use during these computations, might be so large as to make electronic duplication of the brain's activity implausible. Moreover, even were this not the case, the algorithms which regulate the computational activity of the brain might be so marvelously subtle as to frustrate their rediscovery by artificial intelligence researchers for a very long time. Our next task is to examine these possibilities.

The human brain consists of approximately $10^{11}$ neurons, though this estimate is uncertain to within a factor of 10. Neurons typically (though not invariably) communicate by transmitting discrete electrical spikes (action potentials) to a population of follower neurons. As far as is known, the precise amplitude and shape of such a spike, and the precise time of its arrival within an interval of 2 milliseconds or so, are physical details which the nervous system is not able to exploit. This allows one to model each spike as a single information-carrying 'bit' which can be present or absent in a neuron's output stream. We can therefore regard a neuron as producing output information at rate of approximately 100 bits/second. This leads to an estimate of $10^{13}$ bits/second, give or take a factor of 100, for the internal 'bandwidth' of the brain.

The computational activity of individual neurons involves a considerable variety of mechanisms, still very imperfectly understood. Nevertheless, a considerable mass of experimental evidence supports the following general picture. Information is transmitted by a neuron to its follower neurons at inter-neuron junctions called *synapses*. A single neuron can have as many as 10,000 such synaptic inputs, though in some cases many fewer and in other cases as many as 100,000 inputs are known to converge on single neurons. Thus the total number of synapses in the brain can be estimated as $10^{15}$, though this estimate is uncertain by a factor of roughly 100. Input signals transmitted to a neuron (generally chemically) across a synapse trigger a wide variety of reactions. A common effect, and one which seems certain to be of particular importance for the fastest computations performed by the brain, is modulation of the ionic conductivity of the affected neuron's membrane, which either raises the voltage of a portion of its interior (excitation) or lowers this voltage (inhibition). The affected neuron then combines the voltage changes generated by such synaptic effects (after attenuation in space and time in a manner determined by the detailed chemistry and geometry of the neuron and its synapses) and, if the resulting combined (e.g. summed) voltage exceeds a reaction threshold, the neuron generates an output spike, which is then transmitted to all its output synapses.

Other forms of synaptic input are known to have slower but longer-lasting biochemical effects than the ionic effects which probably support the bulk of the brain's information-transmuting activity. Stimulation of certain synapses can, for example, trigger enzymatic activities within a neuron which modify its biosynthetic activities in significant ways, e.g. by increasing or decreasing its susceptibility to subsequent fast excitatory or inhibitory stimuli acting ionically. Depending on the chemical effects involved, such synaptic modification of faster synaptic responses can exert an effect either for relatively short periods (e.g. 50 milliseconds), or for periods of several seconds, minutes, or days; perhaps even permanently. Other synaptically-triggered enzymatic reactions can initiate sequenced biochemical changes which, for example, enhance a neuron's subsequent electrical response for several tens of milliseconds but then inhibit its responses for a longer period, leading to complex patterned alternations of behavior. The varied single-neuron behaviors which can be engendered by the wide spectrum of enzymatic actions that have been demonstrated experimentally have been explored in simple animals such as *Aplysia*, some of whose neurons are known to have highly individualized patterns of continuing, or periodic, or burst activity.

Though it is not easy to summarize such a wide range of synaptic response patterns by a few numbers representing the information processing power and storage capacity of a single neuron, the following estimates do not seem wildly unfair. One byte may well suffice to represent the long-term strength of each of a neuron's synapses with sufficient accuracy. Four additional bytes can then be taken to give a sufficiently complete representation of the short-term biochemical state of both sides of a synapse and of the state of the corresponding synaptic gap, as determined by its stimulation history up to a given moment. Such very rough quantitative guesses lead us to estimate the long-term memory available to the brain as (very roughly) $10^{16}$ bytes, and the amount of shorter-term data needed to characterize the state of each of its synapses as $4 \times 10^{16}$ bytes. The logical activity of each neuron can then be regarded very roughly as a process which combines 10,000 input bytes with roughly 40,000 synapse status bytes 100 times each second; we can guess the amount of (analog) arithmetic required for this to be (again very roughly) $10^7$ elementary arithmetic operations per neuron per second, leading us to estimate the computing rate needed to emulate the entire brain on a neuron-by-neuron basis as roughly $10^{18}$ arithmetic operations per second. (Of course, much lower computation rates might suffice to represent the logical content of the brain's activity, if we could find out what this was.)

Even though it is not inconceivable that the estimates offered in the preceding paragraph might have to be increased by factors by $10^3$ or even $10^4$, it seems much more likely that they overstate the usable arithmetic and memory storage capacity of the brain by large factors. Indeed, anatomical inspection and direct recording of neuronal activity both make it appear that the degree of precision in the wiring of the brain is low, and that (perhaps in consequence) the brain typically employs hundreds of neurons to perform closely similar calculations whose results are then only used in some coarsely averaged manner as mental activity proceeds. Nevertheless, our estimates, $10^{18}$ arithmetic operations per second and $10^{16}$ or $10^{17}$ bytes of memory available, still allow for stupendous amounts of calculation and might therefore represent a very significant obstacle to the easy advance of artificial intelligence. The largest general purpose supercomputers are not likely to attain performance levels of more than $10^{12}$ arithmetic operations per second during the next decade. (However, specially designed systolic arrays might attain higher speeds for particular operations.) Though rotating memories capable of storing $10^{12}$ bytes do not appear entirely infeasible, electronic memories seem entirely unlikely to exceed $10^{12}$ or even $10^{11}$ bytes within a decade. The estimated internal communication bandwidth of the brain, roughly $10^{12}$ bytes/sec., seems somewhat easier to match artificially, e.g. by a switching network of $10^4$ ports each capable of handling 100 megabytes per second.

We conclude that the very largest supercomputer systems likely to be developed over the next decade or two may still fall for short of the raw information-processing capabilities of the brain, perhaps by a factor of $10^6$ or more. However, differences in the algorithmic effectiveness with which this computing power is employed can outweigh even so large a factor. We now turn to discuss this point.

### 3. The Kind of 'Program' which the Brain is Likely to Employ

Algorithms regulating the computational activity of the living brain which are exceptionally effective and which also are subtle enough to defy rediscovery might represent another significant limit to the progress of artificial intelligence. However, it seems unlikely that such algorithms play a role in the functioning of the brain, so that algorithmic considerations seem likely to favor artificial systems over natural systems. This advantage could help artificial systems overcome the substantial advantage in raw computing power which we have ascribed to the brain. The neurological argument which seems to justify such a conclusion is as follows. If for the moment we set all effects of postnatally learned information aside, neuroembryological evidence hints at the following picture of the innate (genetically determined) capabilities of the brain (including its learning capabilities). Within a developing nervous tissue, particular subpopulations of cells take on specialized morphological and biochemical characteristics. (Such cell specialization is of course the basic mechanism of embryological development in general.) Almost nothing is yet known concerning the total number of specialized neuronal subpopulations which develop; however what morphological and physiological evidence there is seems consistent with the assumption that these number several thousands or ten-thousands. Each of these cell populations grows to a genetically determined extent, thereby generating a large or small portion of the nervous system.

Cell migration over large or small distances, and genetically determined temporal sequencing of growth phases among various neuronal subpopulations, also play a role in determining final tissue morphology and neuronal connection patterns. As they specialize, neurons grow thin projections (axons and dendrites) which can extend as little as a few microns or as much as a meter in length. The paths along which these neuronal projections grow seem to be determined by such biochemical factors as the ability of the 'growth cones' present at the tip of a growing axon to react to chemicals present on the surfaces of the cells they touch. These reactions seem to result in selective affinities and adhesions, and to be supplemented by more diffuse chemical gradients present in developing tissue. The phased growth of small spots of tissue to which particular sorts of axonal growth cones have positive or negative affinities can cause these growth cones to move sharply in particular directions, allowing intricately interwoven neuronal morphologies to develop. Once the developing projections from a given neuronal subpopulation have reached their target tissue, similar chemical mechanisms may be used to recognize various subpopulations present in that tissue, and to guide the formation of connections, having specific strengths, among the members of particular ‘immigrant’ and ‘native’ neuron subpopulations. Moreover, if the source tissue sending projections to a target tissue is large enough for its geometric extent to have informational significance, cells within both these areas can be marked, perhaps by chemical gradients, in a manner representing their location, and such markings can then strengthen or weaken the affinities which axons with given origins have for cells at corresponding locations in an extended target tissue. Such geometrically conditioned affinities would allow separate neuronal areas to connect to each other in geometrically regular and informationally significant spatial patterns, which various learning-like postnatal growth processes can then refine.

Basic developmental mechanisms of roughly the kind just sketched seem to define the innate structures present in the brain immediately after birth. The forms of information processing which such structures can most naturally realize can be abstracted approximately as follows. Assume various populations of cells numbered by an index $i$ running from $1$ to $N$. The $i$-th population can consist of many cells, which can be distributed geometrically in 1, 2, or 3 dimensions, and thus can occupy positions correspond to values of a parameter $x$ varying over the interior either of the unit interval, square, or cube, in a manner which we suppose to be continuous. We can then describe the average internal state of type $i$ neurons located at point $x$ by a function $s_i(x)$ whose values are vectors of as many components as are needed to describe this state with sufficient accuracy. In the absence of further excitatory or inhibitory input, this internal state $s_i$ will decay after $dt$ milliseconds to a state $s_i + f_i(s_i)\,dt$. The rate at which tissue $i$ generates output spikes is a function $O_i(s_i)$ of its internal state. If outputs from subpopulation $i$ impinge upon the $j$-th cell population at all, we can describe their effect as follows: information transmitted from population $j$ is received after some characteristic delay $d_{ji}$. A spike train transmitted with intensity $O_i(s_i(x_i))$ from a point $x_i$ in the $i$-th cell population will be received with intensity

$$I_{ji}(x_j) = \int K_{ji}(x_j, x_i)\, O_i(s_i(x_i))dx_i,$$

where the kernel $K_{ji}$ defines the extent of ‘blurring’ which occurs during transmission, as determined by such factors as the affinity which axons originating at place $x$ in population $i$ have for particular points in their target tissue, the extent and typical direction of dendritic and axonal spreading within this tissue, etc. The various inputs of types $i_1, i_2, \cdots i_k$ impinging on a type $j$ neuron at position $x_j$ modify the evolution of its internal state, causing this state to take on the value

$$s_j(x_j) + f_j(s_j(x_j), I_{ji_1}(x_j), \ldots, I_{ji_k}(x_j))\,dt$$

after $dt$ seconds.

Our simplified picture of the patterns of excitation that develop in (untrained) neural tissue is then summed up a system of equations having the integrodifferential form

$$\frac{d}{dt} s_j(x, t) = f_j\,(s_j(x, t), \int K_{ji_1}(x, y)\, O_{i_1}(s_{i_1}(y, t - d_{ji_1})dy, \cdots, \tag{1}$$

$$\int K_{ji_k}(x, y)\, O_{i_k}(s_{i_k}(y, t - d_{ji_k})\,dy), \quad j = 1, \cdots, N.$$

The decay functions $f_j$, delays $d_{ij}$, and kernels $K_{ji}(x, y)$ appearing in this last equation are intended to represent all relevant neuroanatomical and biochemical aspects of a particular neural system in abstract form.

Systems of equations of the form (1) are capable of realizing a great variety of periodic and aperiodic behaviors, including arbitrary Boolean switching actions. Nevertheless, the algorithmic structure of a neural system behaving in a manner describable by a system of such form is significantly restricted, at least if the number $N$ of distinct neuronal species allowed in the model is relatively small (e.g. not more than several thousand) and if the behavior of the system is as stable in regard to fair-sized perturbations of the kernels $K_{ji}$, delays $d_{ji}$, response functions $f_j$, and injections of random noise as mammalian nervous systems seem to be. In particular, it would seem to be impossible for such systems to use any delicately balanced algorithm involving closely coordinated iterated motions of data between intricately arranged storage points.

Better understanding of the manner in which the brain makes use of the enormous mass of learned information which it is able to store will doubtless reveal less constrained ways in which it can deal with massive amounts of information having a discrete character rather than the image-like character of the functions $s_i(x)$ appearing in (1). Nevertheless, combinatorially complex forms of information processing seem unlikely to be used, in particular because they seem unlikely to have arisen in the course of organic evolution, which typically proceeds by progressive adaptation and enlargement of existing structures rather than by sudden leaps. In contrast, artificial data analysis systems can often make enormously effective use of delicately balanced patterns of data processing and motion, which often speed up the generation of needed intermediate or final results by many orders of magnitude. Algorithmic considerations seem therefore to favor artificial systems over natural.

## 4. Limits Set by the Quantitative Theory of Computational Complexity

The abstract model of nervous tissue set out in the preceding section serves more comfortably to generate surmises concerning the operation of sensory functions such as vision, tactile sensation, and hearing than to represent the brain's ability to deal with more discrete or symbolic material, i.e. to reason. The most remarkable, and perhaps fundamental, part of this is the brain's ability to organize information presented in relatively disordered form into internally organized structures on which sophisticated, coherent courses of symbolic and of real-world action can be based. It is the present lack of this ability that makes it necessary to *program* computers rather than simply to *teach* them; teaching would be vastly more convenient and which would bring the era of artificial intelligence very close if it became possible. To clarify this basic distinction, note that the ability of computers to accept, retain, and utilize *fully structured material* is already enormously superhuman, e.g. a computer can acquire and proceed to use the very complex set of rules for compiling a programming language in just a few seconds; nothing in the biological world other than the transmission of a full set of genes during conception matches this enormous rate of information transfer. On the other hand, although a computer can easily acquire and retain the whole text of the *Encyclopedia Brittanica* (even by reading its pages successively) computers are at present incapable of making any active use of the information which these volumes contain, since this text falls far short of the degree of rigorous order and standardization which present computers require. If this basic obstacle could be overcome, computers could immediately proceed to ingest the information contained in all the world's libraries and use this information with superhuman effectiveness. For this reason, a basic goal of artificial intelligence research has been the discovery of principles of self-organization robust enough to apply to a wide variety of information sources. Any such organizing principle would have to allow coherent structures capable of directly guiding some form of computer action to be generated automatically from relatively disorganized, fragmented input.

The present state of artificial intelligence research is most fundamentally characterized by the fact that no such robust principle of self organization is as yet known, even though many possibilities have been tried. Indeed, high hopes for the success of one or another apparently promising general principle of this type have characterized successive periods of research in the history of the subject. A typical attempt of this kind, particularly intriguing because of the great generality and potential power of the mathematical tools which it proposes to employ, has been the attempt to use formalisms drawn from symbolic logic as the basis for a self-organization capability. Mathematical axioms and theorems are mutually consistent fragments of information which can be accumulated separately and indefinitely; mathematical proofs based on these axioms and theorems are highly structured wholes which arise from these fragments according to the simple, well-understood principles of formal logic. If they could be generated automatically, these proofs, or various proof-like structures easily derivable from them, could be used almost immediately to produce many other symbolic structures, including computer programs. Here a door to the most ambitious goals of artificial intelligence seems to swing open. Unfortunately, this prospect, like all others that have been explored to date, has proved to be blocked by fundamental considerations of computational efficiency, which we will now review.

The modern quantitative theory of computational infeasibility deriving from the work of Godel and Church allows one to prove rigorously that enormous computational costs will always make it impossible for programmed systems to answer certain general classes of questions in all cases. The original Church-Godel result is qualitative rather than quantitative, and can be summed up in a short *unsolvability* statement: there can exist no computer program $P$ which is capable of examining every other program $Q$ and determining correctly, in finite time, whether $Q$ will run forever or halt eventually. Since many other combinatorial problems can easily be proved equivalent in difficulty to this basic unsolvable problem, they are just as unsolvable. Recent more quantitative work along these same lines has shown that there exist significant classes of mathematical problems which, although algorithmically solvable in the sense that one can write programs capable of solving each of the problems in such a class, are nevertheless *intractable*, since most of the problems in each of these classes carry minimal computational costs which rise with enormous rapidity as the program classes are progressively generalized in directions which eventually carry them over into the Church-Godel zone of complete unsolvability. As this happens, seemingly small loosenings of the constraints defining a particular class of problems always increase the cost of dealing with the generalized class enormously.

Problems in computational logic, whose efficient solution would provide very general and powerful tools for development of artificial intelligence, illustrate these general remarks. Any mathematical statement can be written in a convenient yet perfectly rigorous way using the simple notations of predicate logic. For example, the predicate statement

$$(\mathit{FOR\ ALL}\ x, y, z, u, v, w) \tag{2}$$

$$(\mathit{REAL}\ (x)\ \&\ \mathit{REAL}\ (y)\ \&\ \mathit{REAL}\ (z)\ \&\ \mathit{REAL}\ (u)\ \&\ \mathit{REAL}\ (v)\ \&\ \mathit{REAL}\ (w)$$

$$\mathit{implies}$$

$$((x + u)^2 + (y + v)^2 + (z + w)^2)^{1/2} \leq (x^2 + y^2 + z^2)^{1/2} + (u^2 + v^2 + w^2)^{1/2})$$

captures the geometric fact that a broken line in three dimensional space is always at least as long as a straight line connecting the same endpoints. (In the preceding formula, clauses of the form $\mathit{REAL}\ (x)$ express the fact that the variable $x$ designates a real number.) Because of their great generality, predicate formalisms like that seen in the preceding formula provide very interesting testing grounds for artificial intelligence research. Any method which allowed the truth or falsity of large classes of formalized statements of this kind to be decided automatically and efficiently would also allow one to perform many other operations, including the automatic composition of many kinds of computer programs, the planning of grasping positions and motions for robot arms, and many many other geometric and spatial analyses. However, a considerable body of rigorous theoretical analysis now rules out this possibility. Specifically, it has been shown that algorithms for deciding the truth of entirely general predicate statements cannot exist, nor can there exist algorithms capable of performing any entirely general process of formal reasoning, construction, or problem solving equivalent in difficulty to the task of classifying entirely general predicate statements as true or false. Indeed, the existence of such algorithms is directly ruled out by the basic Church-Godel theorem referenced above. On the other hand, algorithms capable of deciding narrower but still quite interesting subclasses of predicate statements do exist. For example, a famous theorem of Tarski asserts the existence of an algorithm capable of deciding any statement concerning real numbers which can be written using only the four elementary arithmetic operations of (addition, subtraction, multiplication, and division), comparisons between real numbers (e.g. clauses of the form 'x is greater than y'), the elementary Boolean connectives (and, or, implies, not), and the standard predicate quantifiers (*FOR ALL x*, *FOR SOME x*). However, the task which this algorithm accomplishes lies close enough to the Church-Godel zone of unsolvability that even apparently slight generalizations of this problem prove to be algorithmically unsolvable. For example, the same decision problem for the class of statements having exactly the same structure, but in which variables designate whole numbers (integers) rather than arbitrary real numbers (which for technical reasons are somewhat easier to deal with), is unsolvable.

Moreover, since the Tarski decision problem for real arithmetic is nearly unsolvable, any algorithm capable of deciding the truth/falsity of any statement of the form described must require enormous, and indeed prohibitive, computational resources in the worst case. Specifically, a theorem of Ferrante and Rackoff, proved in 1975 shows that the running time even of the fastest possible algorithm capable of deciding the truth or falsity of every statement *s* of Tarski form must rise exponentially with the length of *s*, for some (though not for all) such statements *s*. Thus in unfavorable cases the minimum running time of such algorithms will be probably in excess of billions of years, making their existence a matter of theoretical interest rather than of practical significance. Theorems of this same sort apply to many other classes of mathematical statements having decision problems of roughly the same degree of inherent difficulty as the Tarski class, and imply even higher degrees of computational difficulty for more general statement classes. For example, although the full class of statements of Tarski form becomes undecidable if applied to integers rather than real numbers, the subclass of statements involving only arithmetic addition, subtraction, and comparison operations (but no multiplications or divisions) remains decidable even if applied to integers. However, here again we lie close enough to the zone of absolute unsolvability for computational costs to rise prohibitively high. More specifically, a theorem of Fisher and Rabin (1974) shows that these costs must be just as large as the Tarski case costs described above.

These general statements of computational infeasibility play the same role in computer science generally and artificial intelligence particularly that the first and second laws of thermodynamics play in physics and engineering, i.e. they set limits to what it is reasonable to attempt. While they do not at all rule out the possibility of artificial intelligence, they do suggest that it cannot be attained by programming any unitary mechanism of complete generality from which all that is needed will follow by simple specialization. Instead, it may be necessary to develop a relatively large number of artificial systems which mimic particular types of reasoning and mental functions in cases specialized enough to admit of particularly efficient treatment, and by systems whose 'coverage', while broad enough to be very useful, is less comprehensive than is assumed by naive mathematical statements of the problems they address. The individual functions thereby produced would then have to be integrated into a software structure capable of a very advanced level of function, which hopefully would also assist substantially in its own further development. Painfully detailed manual development of very many separate subcomponents of a highly complex total system capable of exhibiting a high level of intelligent function will only be avoided if some relatively uniform principle allowing computers to learn in human-like fashion is somehow developed. At present we have no real inkling of how this might be done, though the preceding model of neural function suggests that it ought somehow to be possible. It is equally unknown whether this present incapacity is a consequence of grossly insufficient computing power, as some of the estimates made earlier in this article seem to suggest, or simply reflects the fact that we have not yet found those simple yet efficient mechanical learning techniques whose discovery will enable much more rapid advance.

## 5. Limitations of the Present State of Knowledge in Artificial Intelligence

Since principles of self-organization allowing generation of broadly useful symbolic structures from more disorganized and fragmentary input would be crucial to the progress of artificial intelligence, work aiming at the discovery of such principles has been much emphasized. Signs of progress in this direction have always generated particular excitement. Unfortunately, all such efforts to date have run aground on the computational cost difficulties outlined in the preceding section. This fundamental fact constrains the immediate perspectives of the field severely. Of course, the many intriguing techniques developed during twenty years of artificial intelligence research do not lack application; indeed, their applications can be expected to grow steadily in scope and number. However, in the absence of any unifying principle of self-organization, these applications must be seen as adaptations of diverse ideas, rather than as systematic accomplishments of a still mythical 'A.I. technology'. We are still at the point at which the success of such applications depends far more on clever special algorithms and code reflecting particular application content than on use of the still impoverished general-purpose tools of artificial intelligence. Moreover, since specialization is still generally vital to success, it is hard to characterize the extent to which success in any one application should be read as representing advance of the artificial intelligence field as a whole: to the degree that an application comes to depend on special techniques, special data layouts, and special algorithmic approaches, we can no longer rightly regard it as evidence for the viability of a general approach distinguishable from artful programming in general. Nevertheless, some of the more specialized research efforts inspired by general artificial-intelligence notions have succeeded modestly in mimicking limited but interesting aspects of mental capabilities such as vision and natural language understanding.

To clarify this assessment, the present status of work along various significant lines will be summarized in this section. It is useful to arrange this work under three main headings: sensory functions, motor control, and reasoning. More detailed articles on the various areas reviewed should also be consulted.

### 5.1. Sensory Functions

These include analysis of images (computer vision), analysis of natural language made available in written form, and of continuous speech.

### 5.1.1. Analysis of Images

In spite of a great deal of work on the first steps of image processing (e.g. deblurring, edge detection) we are still far from being able to duplicate the eye's remarkable ability to detect objects in the presence of large amounts of visual disguise. Nevertheless our ability to identify objects within scenes is steadily improving, particularly for scenes containing only objects whose geometry and coloration is known in advance. Even if large parts of the objects present are obscured, such scenes can be handled more easily than entirely general images (e.g. images of outdoor scenes containing shrubbery.) This reflects the fact that the problem of identifying known bodies and determining their orientation (the 'model based' vision problem) is entirely objective; in contrast, the problem of imposing useful perceptual groupings on entirely general scenes is at least partly psychological, i.e. to solve this second problem we need to match the functions of the human visual system well enough for introspection to serve as an accurate guide to the way in which a robot vision system will react to a scene.

Among the many methods which are becoming available for handling the easier 'model based' vision problem are: direct matching of curves having fixed geometric position on known object surfaces; use of projective invariants of object silhouettes; probing techniques applicable for objects known to be presented in one of a finite number of allowed positions (e.g. objects lying on a table-top or conveyor belt) or on which one or more characteristic features can be reliably located; geometric reasoning using features (such as corners, straight corners, straight edges, circles) which can be detected directly or by statistical (e.g. Hough transform) methods.

Another promising object recognition technique is computation of invariants of local shape (rotational invariants) for the edges of two-dimensional figures and for the 'ridges' (curves along which at least one of a surface's extrinsic curvatures is large) of 3-dimensional objects. Any sharp color or reflectivity boundaries present on the surfaces of (painted or otherwise marked) 3-dimensional objects can also be used. To the extent that is is possible to define invariants stable against the disturbing effects of observational noise, changes in illumination level, viewing angle, specularity, etc., this technique can support recognition even of heavily obscured objects and allows use of hashing techniques which greatly reduce the cost of identifying objects selected from large vocabularies of potential candidates. Beyond this, sophisticated use of color and texture cues available on object surfaces may prove possible. Here, however, we come to the point at which the human (or mammalian) visual system displays a sophistication that researchers seem far from being able to match, even after several decades of determined effort. In some remarkable way, the eye is able to integrate the evidential weight of fragmentary clues, and to make use not only of dotted and dashed lines but of computationally elusive texture boundaries, vague differences of shading, and curves which are very badly broken up by obscuring objects (e.g. foliage) and complex shadow patterns. All this can be done in a manner resistant to the confusing effects of very large changes in illumination pattern, intense specularities, image blurring, and the myriad other effects all too painfully familiar to the vision researcher. Finally, all this is possible for scenes containing large numbers of objects, some unfamiliar, seen in a great variety of apparent sizes, from sharp and severely distorting angles, and in the absence of binocular information.

At the present time we have little understanding of how all of this is accomplished, and at what computational cost.

However, it is clear that image processing tends to be very expensive computationally (e.g. initial analysis of an image often requires examination of between 250,000 and 1,000,000 separate image pixels), so that substantially faster processors than are now available may prove to assist the development of this very challenging subject. These processors may include special purpose chips able to apply basic image analysis operations at high speed.

Robot systems equipped with tactile sensors acquire 'tactile images' which are of much lower resolution than visual images but can be analyzed using techniques like those applicable to visual images.

### 5.1.2. Recognition of Continuous Speech

The ability to interpret continuous speech, i.e. to hear continuously varying soundwave patterns generated by speakers of a familiar language and to transform them into roughly equivalent written sequences of phoneme indicators (or into standard word spellings) is a basic capability of the human auditory and nervous system, shared to some extent with an enormous range of other living creatures, e.g. birds sensitive to particular birdsongs. The history of efforts to give computers a comparable ability provides a nice illustration of the possibilities and difficulties facing artificial intelligence research focused on sensory areas.

Processing of speech begins with spectral analysis of an impinging sound system to extract energy intensities in a range of frequency channels. These intensities define a family of physical parameters of the impinging speech signal which vary continuously through time, and hence allow the received signal to be regarded as a continuous curve $C_o(t)$ in $n_o$-dimensional space, where $n_o$ (which typically has a value lying somewhere in the range of 5 to 20) is the number of distinct energy intensities (or other physical parameters of the incoming sound) extracted. These initial parameters can then be supplemented by adding various derivatives, smoothed derivatives, or other locally defined time-invariant functionals as additional parameters, to produce a modified continuous curve $C_1(t)$, having a somewhat larger number $n_1$ of parameters, as an improved description of the incoming signal. This description can in turn be subjected to an appropriate nonlinear transformation to normalize it for such speaker-dependent variables as pitch of voice and speech rate. This yields a parametrized multi-dimensional curve $C(t)$ suitable as input to the next, more symbolic, steps of processing.

The necessary transition to a symbolic stage of processing can be accomplished in a variety of ways. A typical technique is to divide the $n$-dimensional space $E^n$ through which the curve $C(t)$ runs into a collection of overlapping regions $R_1,....,R_m$, each of which corresponds to one of the basic phonemes $p_j$ recognized by the language to which an utterance belongs. Passage of the curve $C$ through a region $R_j$ is then regarded as indication that the corresponding phoneme $p_j$ has been pronounced. Since the regions $R_j$ can overlap, the specific phoneme being pronounced (or, more properly, 'heard' at any given moment) is somewhat ambiguous. (Instead of phonemes, the basic symbols into which $C(t)$ is (ambiguously) converted can be larger speech units, e.g. full syllables, or 'demisyllables' consisting of a consonant preceding or following a vowel or phoneme fragment.) Numerical probabilities for the presence of any given phoneme (or other primitive symbolic element) can also be computed, e.g. by using a smoothly varying function $f_j$ positive only within $R_j$, and then forming $f_j(C(t))$ instead of the simpler Boolean quantity $C(t) \in R_j$.

This converts the incoming acoustic signal into a sequence of the form

$$\cdots \{s_1^{(1)}, \cdots ,s_{k_1}^{(1)}\},\{s_1^{(2)}, \cdots ,s_{k_2}^{(2)}\}, \cdots \{s_1^{(p)}, \cdots ,s_{k_p}^{(p)}\}, \cdots \tag{3}$$

whose successive elements are sets of phoneme (or syllable, or demisyllable) symbols $s_j^{\,i}$. The members of each such set represent all the phonemes which designate sounds close enough to the incoming signal during a specific instant of time that they might have been pronounced during that instant. What remains is to disambiguate the sequence into a final perceived phoneme string

$$\cdots s_{2_1}^{(1)} \; s_{i_2}^{(2)} \cdots s_{i_p}^{(p)\cdots} \tag{4}$$

each of whose successive symbols $\cdots s_{i_j}^{(j)\cdots}$ belongs to the corresponding set in the sequence (3). As cleverly pointed out by John Cocke, this is like the problem of decoding an English language message that has been ambiguously spelled out by dialing it on a standard telephone dial and by transmitting the resulting digits only (note that each digit transmitted then refers ambiguously to one the three possible associated letters). Such disambiguation must of course rest on other knowledge concerning the phoneme (or syllable, or demisyllable) sequences that can legitimately occur in the language to which the expected utterance belongs. Several approaches to this goal are possible:

(i) One can use some form of (possibly multi-level) grammar to define the set of all word sequences, and from this the set of all syllable, demisyllable, and phoneme sequences which are legal (or likely to occur) in the language of the utterance being analyzed. The computational problem then becomes that of finding the grammatically valid phoneme sequence sequences (4) consistent with the ambiguous input sequence descriptor (3).

(ii) One can proceed (at least for some of the phonemic or syllabic levels that would otherwise have to be described by formal grammars) in purely statistical fashion. This can be done by regarding utterances in the language to be analyzed as outputs from a Markov source, whose characteristics can be ascertained by collecting data on the frequency with which a given phoneme follows a preceding sequence of one, two, or more known phonemes. Then the most acceptable interpretation (4) of the ambiguous input sequence (3) can be defined as the most probable sequence consistent with (3), and can be calculated by some dynamic programming procedure, e.g. the Viterbi algorithm. A probabilistic approach of this kind can make good use of numerical measures of likelihood associated with the various alternatives appearing in each of the sequences constituting (3).

Substantial research efforts mounted during the last few years have significantly increased the speed and robustness of interpretation techniques of the kind just described. VLSI chips able to accomplish the initial (analog) steps of processing (spectral decomposition, signal filtering and differentiation, signal normalization) and perhaps even generation of the first level phoneme stream (3) should soon be available.

The interpretation problem is eased substantially for words spoken in isolation, since in this case direct matching techniques able to span a word's whole extent are computationally feasible; devices capable of recognizing vocabularies of several hundred words spoken in isolation are already available commercially. For continuous speech, the still very onerous computational cost of disambiguating (3) into (4) has forced researchers to concentrate much of their attention on various heuristic schemes for reducing this cost. However, neither powerful algorithms for accomplishing this efficiently, nor any real analysis of the inherent computational difficulty of the problem is yet available. Techniques applied have ranged from systematic use of probabilistic techniques or relaxation-labeling ideas to entirely *ad hoc* schemes for combining clues detected by multiple interpretation processes acting at numerous phonemic, syntactic, and even semantic levels.

One is entitled to feel a certain optimism concerning continued progress of work in this area, since the inherent time-sequence of the signals being analyzed assists powerfully, as do the greatly increased levels of computational power made available by VLSI technology. Perhaps the overall theme that emerges here is the accessibility of the simpler human input modalities to automation, given sufficiently large increases in computational capability.

### 5.1.3. Analysis of Natural Language

Simple formal grammars (e.g. context free grammars) of the kind used to define the structure of programming languages serve remarkably well to define the basic structure of natural language syntax. However, natural language admits a far greater variety of syntactic irregularities, special usages and idioms, fragmentary and semi-grammatical usages, and specialized sublanguages and jargons such as doctor's English, criminal argot, and Jivetalk. Compared to artificial languages, natural language appears as an overgrown jungle whose effective description, even at the purely syntactic level, requires grammars whose symbols carry many kinds of attributes (e.g. 'count noun', 'animate noun'), treat many words in special ways, and require elaborate, sometimes explicitly procedural, handling. Natural language also involves many syntactic and semantic ambiguities which can only be resolved using extensive real-world knowledge, e.g. 'Standing in the pen, the cattleman took his pen from his pocket'. In spite of decades of work on computational linguistics, we are still far from possessing any computerized natural language analysis system that can either deal with the very wide range of phenomena (especially errors and sentence fragments) appearing in informal English, or handle more than a very few of the specialized sublanguages with which people deal commonly or comfortably, or resolve ambiguities at all well.

Attempts to treat the semantics of natural language automatically confront artificial intelligence research with problems far deeper and seemingly less tractable than those of syntactic analysis. A language's semantics imbeds its set of grammatical sentences in a deductive framework, making it possible to use the overt text of a discourse to deduce facts not explicit in this text. Moreover, certain combinations of grammatical sentences will then be semantically inconsistent, allowing certain otherwise ambiguous sentences to be disambiguated on semantic grounds. For example, without semantics the sentence 'I noticed a man on the road wearing a dark hat' would admit an analysis in which the road, rather than the man, was wearing the hat, as in 'I noticed a man on the road leading to the North end of town'. Semantic relationships allow resolution of many other ambiguities which natural language syntax allows, e.g. ambiguities of quantifier ordering ('A woman gives birth in the U.S. every 5 minutes') and anaphora ('John bought his groceries in several adjoining small shops. They cost 20 dollars.')

Any fully satisfactory formalization of the semantics of natural language must address all of the following very challenging problems, plus others.

(1) A deductive framework accommodating a wide variety of probabilistic and other informal arguments going far beyond the kinds of rigorous deduction allowed in mathematics must be provided. Among other things, one needs to allow controlled relaxation of normal semantic restrictions in order to accommodate unusual sentences like 'The long road and the slender tree sat around the wizard's table talking. The road was wearing a dark brown hat' in texts recognized as fairy stories, even though roads wearing hats are ordinarily disallowed semantically.

(2) The framework chosen must accommodate the whole enormous range of facts entering into natural language discourse, including all the common sense facts of naive physics concerning such categories as 'above' and 'below', 'inside' and 'outside', ' big' and 'little', etc. Means for reasoning about such elusive matters as plans, knowledge, beliefs, and motives must also be provided, to say nothing of social phenomena such as embarrassment.

(3) Inference within a semantic framework must generally be quite efficient, e.g. so that fast inferences can be used to disambiguate syntactically ambiguous sentences and/or to resolve anaphoric references in lengthy

text streams.

At present we have little idea of how to treat most of these issues, which collectively reach to the heart of the artificial intelligence enterprise. For example, no 'probabilistic' or 'fuzzy' formalism beyond the well-defined but rigid semantic area mapped out by propositional and predicate logic has as yet demonstrated advantages sufficient to win it general acceptance. Moreover, the basic problem of what primitives a semantic formalism should use is surrounded by deep and ill-fathomed questions. One possibility is to somehow simplify the capture of information concerning the very many concepts appearing in natural language discourse by re-expressing them in terms of some much smaller family of simpler primitives whose properties can then be expressed by a significantly smaller set of rules. (This simplification would in effect require finding some way of extending the analytic reductionism characteristic of theoretical science to the entire range of phenomena which natural discourse addresses.) Any expectation that this can succeed easily is discouraged by consideration of the slow pace with which science has previously advanced into entirely new fields, and on the enormous computations sometimes required to apply general scientific laws to particular concrete cases. The opposite approach is to somehow build a semantic formalism which handles the very many terms appearing in natural language as unanalyzed primitives which it relates to each other by comprehensive sets of axiom-like formulae. Belief that this approach can succeed easily or rapidly is discouraged by the formidable difficulties of steering proofs in predicate calculus systems that try to deal with more than a dozen or so carefully crafted axioms.

Measured against these deeply rooted problems, existing techniques for dealing with natural language semantics appear sketchy indeed. *Semantic network* systems attempt to organize the enormous variety of objects and predicates appearing in ordinary discourse by representing them as nodes in graphs whose edges represent various logical relationships which are felt to be particularly fundamental to common elementary inferences. For example, such edges may connect nouns A and B whenever A is a 'kind of' B (e.g. when A is 'man' and 'B' is 'mammal') or when A is a 'part of' B (e.g. when A is 'arm' and B is 'man'.) A second aim of schemes of this sort is to accelerate simple semantic deductions by making the information they require directly available through short chains of pointers and by grouping related information needed for the commonest types of deduction under appropriate headings. The feasibility of attempts of this kind could only be demonstrated by exhibiting at least one readily extensible system able to cover some extensive domain of practical knowledge robustly, something which no one has yet done successfully.

Roger Shank's 1977 'conceptual dependency' scheme represents an attempt to reduce the myriad elements appearing in ordinary discourse to a much smaller set of semantic subcategories. It is not inconceivable that such an attempt should yield some useful degree of systematization, even though a pessimist might might view it as a futile effort to enlarge the applicability of scientific modeling by casual invention of a classification scheme. The categories proposed by Shank include 'acts' (essentially verbs, which it is proposed to further subdivide as variants of purported primitive acts such as 'propel', 'ingest', 'expel', 'speak', etc.), 'picture producers' (essentially nouns), 'times', 'locations', etc. A related aim here is to classify all the inferences which attach to entities of these proposed semantic categories.

Marvin Minsky's 'frames' and the associated 'scripts' proposed by Shank define a more general (but accordingly more empty) framework for organizing common sense knowledge in a stereotyped form. Minsky proposes to classify all the logical entities (e.g. nouns) that can appear in a semantic network system into (a possibly large number of) fixed categories. With each such category, a Minsky 'frame' associates a fixed-format record layout listing all the attributes which an item of the given category might have, together with all the values or categories of values which each particular attribute can assume. For example, the frame for entities of category 'restaurant' might have a 'type' field with possible values 'cafeteria', 'full-service', 'full-service-with-hostess', etc., a 'food-style' field with possible values including 'Fish-and-chips', 'Mexican', 'Chinese', 'Thai', 'Seafood', and so forth. Categories can be defined to be specializations of more encompassing categories, whose attributes they inherit; certain of the attributes of a category can be optional.

Shank proposes to include records of another fundamental kind called 'scripts' in semantic systems. These are to be used to describe categories of activity (rather than of objects, as with 'frames'). Basically they list sequences of subactivities, which can in principle be conditional on specified conditions. 'Frames' and 'Scripts' are tied together by the fact that a script can specify the kinds of objects expected to appear in the activities it describes (by including pointers to the corresponding frames), while the frames describing an entity type can reference scripts describing the activities typically associated with these entities.

Taken *per se*, this mechanism is little more than a way of organizing some aspects of the data with which full-fledged semantic inference systems will have to deal, and does not answer the questions of how such an inference system is to be created any more than the inclusion of vaguely similar record types in programming languages such as Pascal and PL/1 answers the question of how to write complex compilers or symbolic manipulation systems using these languages. However, it can also be read as suggesting a semantic interpretation scheme having something of a 'higher level syntax' flavor. Specifically, Shank's 'scripts' can be viewed as higher level grammars defining a language of semantically plausible sentence sequences (whose rudimentary elements are clauses or other sentence fragments, already preparsed in some more standard syntactic sense). This 'grammar' of scripts would allow much 'nulling' of script elements, but then by using such a grammar to 'parse' a text and immediately 'unparsing' the result, with element nulling forbidden, one can hope to make explicit certain simple but very useful classes of normally implicit inferred elements. (Since grammars which allow large amounts of nulling tend to interpret given texts in highly ambiguous fashion, application of a scheme of the sort described may depend upon a rule which prefers the 'shortest' or 'simplest' semantic script-parse of a text to all others. Such a rule would amount to requiring that only those implicit elements necessary to a text's semantic interpretation could rightfully be inferred. Alternatively, the scripts driving the semantic interpretation process could associate probabilities with each elementary interpretation step, and some rule defining 'most probable' interpretations could be used.) A 'grammar of scripts' used in this way will necessarily be context dependent, since semantic connections would have to be maintained between elements (e.g. explicit or implicit nouns or pronouns) recognized at one point of a text and matching occurrences elsewhere. Hence 'parsing' according to such a grammar might come to resemble the very inefficient processes of computational logic much more than the relatively efficient processes of ordinary syntactic analysis.

It would however be easier to take such rationalizing suggestions seriously if straightforward formalisms had been proposed for use in this area and if some initial analysis of their computational cost were available. Unfortunately, however, the literature contains little but preliminary and often confusing heuristic suggestions and computational schemes set out without much justification, no one of which seems to have gained any general degree of acceptance.

This brief review of the difficulties which confront attempts to automate natural language understanding underscores the wisdom of Turing's 1950 suggestion that ability to conduct natural-seeming conversations should be regarded as a touchstone of progress in artificial intelligence. In spite of much work, even a computer able to read simple stories (e.g. ordinary children's stories or newspaper articles) and to answer simple questions about their content still lies far beyond us. Existing semantic analysis systems are fragile laboratory constructions which can deal only with narrowly restricted subject domains. The mechanisms thus far suggested as bases for more comprehensive semantic systems are all quite primitive. Since the problems with which they must deal seem to encompass almost the whole subject matter of artificial intelligence, only slow progress can be predicted.

## 5.2. Motor Control, Modeling of Spatial Environments, Motion Planning

Our review of these topics will illustrate the point that areas of artificial intelligence to which classical scientific and algorithmic techniques apply can be expected to progress more rapidly than areas which deal with deeper problems for which only less focused approaches are available. Many of the capabilities reviewed in this section are being explored in connection with industrial robotics. Since many of the problems encountered are technical rather than fundamental, it is reasonable to expect steady progress, at a rate largely determined by the resources brought to bear. However, it should be noted that work in this area creates very challenging problems of software systems integration, involves a complex mix of technologies, and is quite expensive. Studies in other areas of artificial intelligence such as computer vision may raise similar practical problems as they advance toward maturity.

Research in motor control aims to devise robots capable of exerting sophisticated hybrid force and positional control over grasped objects and to construct robots which can walk, run, leap, and climb. Typical problems of manipulation are to tie a knot in rope, to thread a nut of imprecisely known shape and pitch onto a bolt, and to pick up a jumbled sheet of cloth and fold it neatly. Techniques adapted from concepts presently belonging to nonlinear control theory (which should be considerably enriched by contact with robotics) should make sophisticated manipulation of rigid objects possible during the next few years. To do this, much work on such classical topics as the frictional and elastic reactions of bodies in contact will be required. Dynamic robot control, such as is involved in walking or running, should also progress steadily over the next few years. However, this will require close study of the complex physical situations created as motor-actuated mechanisms having various geometries and dynamic behaviors enter into repetitive contact with supporting surfaces.

The problems of dealing with nonrigid objects, e.g. cloth, are much less understood, and we lack even a vocabulary for describing some of the basic operations involved. How, for example, is a robot to find the edges of a hanging sheet of cloth preparatory to folding it? Roboticists have not yet begun to grapple seriously with such problems, and it is not now understood whether these will permit of uniform attacks or require development of special analyses and approaches in a large number of different cases.

With a few experimental exceptions, today's robots do not maintain any systematic internal model of their environment; the environment is typically known to them only as a source of tactile or visual interrupts, all sense of external object identity being lost as soon as a grasped object is set down or passes out of sight. To develop any deeper understanding of the environment, robots will require far more sophisticated environment-modeling software than is now available. Although the basic principles require for this are largely available from classical physics and geometry, it remains a considerable challenge to devise algorithms capable of performing the required computations with acceptable efficiency. For example, even though the fields of computational geometry and geometric modeling have developed vigorously, we still lack algorithms able to perform such basic operations as detecting intersections between curved surfaces rapidly. More sophisticated modeling operations are needed, e.g. simulation of the paths along which one model object will roll or slide along a given surface, and of the frictional or other forces involved in such motions. These raise yet another range of problems directly significant for artificial intelligence, but which are bound to tax the best efforts of numerical analysts, geometers, and students of mechanics. Doubtless much can be done here, but there is little reason why these problems will advance more rapidly when viewed as problems of artificial intelligence than they would when viewed as problems in geometry and mechanics. In particular, although some artificial intelligence researchers have hoped to construct a semi-symbolic 'naive physics' which could calculate the qualitative outcome of common interactions between physical bodies more cheaply than is possible by detailed physical/geometric modeling, this idea is still in altogether too rudimentary a state for fast success to be likely.

Considerable attention has focused recently on the problem of *motion planning* for robot-controlled bodies moving in obstacle-filled environments. The problem here is to determine whether one or more objects of known shape, moving in an environment containing obstacles of other known shapes, can pass from one specified position to another without colliding either with the obstacles or with each other. In variants of this problem, the obstacles may be moving and the controlled objects constrained to move at bounded rates or with bounded accelerations; or the geometry of the obstacles may be known only in part (but then sensors able to detect object proximity must be available); or it may be required to calculate shortest, or fastest, or most energy-efficient paths. Recent work along geometric lines has begun to elucidate this circle of problems, but doing so has required development of steadily more subtle algorithms drawing heavily on the computational geometer's bag of tricks. This is clearly an area of artificial intelligence research which has advanced by moving closer to other more traditional areas of science, which suggests that at least for the present it may also be easier for other branches of artificial intelligence research to progress in this relatively conservative fashion than by relying on the seemingly more general, but often more vacuous, symbolic search methods traditionally associated with the artificial intelligence field.

### 5.3. Reasoning, Planning, Knowledge Representation, Expert Systems

Workers in artificial intelligence have explored many formal schemes which promised to produce useful structures automatically from less structured input. These have included graph search, the predicate logic mechanisms reviewed earlier, 'rule-based' systems, and the sequencing schemes used as 'inference engines' in expert systems. The most common methods of this sort will be revieweiewedd in the following paragraphs. Attempts to apply any of these schemes wholesale have invariably been defeated by the same combinatorial explosion which makes universal application of predicate logic techniques infeasible.

### 5.3.1. Graph Search

Many problems can be reformulated as that of finding a path between two known points within a graph. Planning and manipulation problems, both physical and symbolic, illustrate this. Such problems are described by defining (1) an initial condition with which manipulation must begin, (2) some target state or states that one aims to reach, and (3) a family of transformations that determines how one can step from state to state.

The problem of chemical synthesis is an example: the target is a compound to be synthesized, the initial state is that in which easily available starting substances are at hand, and the allowed manipulations are the elementary reactions known to the chemist. The problem of symbolic integration is a second example: some initially given formula $F$ containing an integral sign defines the starting state, any formula mathematically equivalent to $F$ but not containing an integral sign is an acceptable target, and the transformations are those that calculus allows.

In all such problems, the collection of available transformations is a heap of relatively independent items which can be expanded freely. Hence the construction of a path through the graph defined by a collection of transformations does represent a situation in which structured entities, namely paths, arise via simple and uniform rules from something unstructured, namely collections of transformations. Early in the history of artificial intelligence it was hoped that this construction could serve as a universal principle of self-organization. However, subsequent experience has repeatedly shown that the size of the graphs needed to represent significant problems in this way can be astronomical, making brute-force search infeasible. To do better, some form of 'guided' or 'pruned' search must be used. 'Guided' search might involve use of some auxiliary heuristic scoring mechanism able to predict the distance to a desired target fairly accurately without the precise path to the target being known. Another possibility is to generate some not fully accurate 'roughed out' preliminary path or plan, and then to try to produce a fully valid graph path by using this rough plan for guidance.

No method for making either of these techniques work at all robustly has yet been developed. A perfectly accurate means of calculating the distance between an arbitrary graph node $g$ and a desired target node $t$ is mathematically equivalent to an algorithm for finding the shortest path to $t$

from any such $g$. Hence we can hardly expect such functions to be available except for problems specialized enough to be subject to complete mathematical analysis *a priori*. Experience seems to show that human attack on substantial problems, especially in problem domains that are at all familiar, involves reaction to so extensive a range of problem and context features as to bar capture by any straightforward scoring heuristic. Guidance by use of rough preliminary plans is frustrated by the present inability of computers to use any adequate notion of similarity in combinatorial domains. In addition, the fact that the number of transformations potentially available, and hence the probability of having to search an exploding number of formal possibilities, tend to rise rapidly once partial solutions and means for amending them are allowed into a problem context.

Pruned search involves either the use of problem symmetries to prevent wasteful exploration of graph paths which have already been searched in some equivalent form, or use of auxiliary rules able to predict that a given graph edge need never be traversed because no path involving this edge can reach the desired target node. Although use of such ideas has proved useful, intractable combinatorial searches generally remain even after such notions are applied, except in particularly fortunate cases whose treatment amounts more to the use of special high-efficiency algorithms than to application of any very general 'artificial intelligence' approach. Moreover, because of the featureless generality of graph-theoretic notions, the formulation of such problems in graph-theoretic terms tends to conceal rather than to reveal opportunities for search pruning.

For all of these reasons, belief in the efficacy of entirely general graph-search approaches has largely disappeared among artificial intelligence researchers, even though graph-based techniques continue to be valued for their generality.

Computer-managed *planning* in artificial intelligence contexts is generally accomplished by reduction to some type of explicit or implicit graph search. The computer maintains internal models of the various situations ('states') that would arise as the result of its tentatively planned actions. These 'states' are treated as the nodes of a graph, whose edges are the actions that could lead from state to state. Since a path through such a graph has then an obvious interpretation as a planned sequence of actions, plans can be generated by specifying an initial and a final state (or by specifying attributes which define an acceptable final state), and by finding a path connecting these two states. As in all graph-theoretic situations this method works well if the graph that needs to be searched is relatively small (e.g. consists of no more than a few thousand nodes). For example, all sorts of simple 'monkey-and-bananas' puzzles can easily be solved by this method. On the other hand, application of this method to more serious planning problems is often infeasible because the graphs involved (explicitly or implicitly) are enormous.

(As an example of this, consider the simple 'nines puzzle', which consists of 8 square pieces in a 3 × 3 frame to be moved between specified configurations. Here the graph of states consists of 9! or 60,480, so even for so simple a problem brute-force graph search begins to become taxing. For the corresponding 4 × 4 puzzle, whose state space involves 16! or over $10^{12}$ nodes, it is completely infeasible.)

### 5.3.2. Predicate Systems

Attempts to generate proofs from collections of mathematical axioms and lemmas by systematic transformation of sets of formalized statements can be regarded as a specialized form of graph search. This is a domain in which heuristic guidance techniques (e.g. rules favoring short formulae over long, or formulae differing little from a target formula $F$ over formulae very different from $F$), problem symmetries, and search-pruning methods have been very extensively explored. Among these are:

(a) the basic *resolution* technique, which handles instantiation of the variables in a set of predicate clauses efficiently by making only those substitutions which arise from some clash between elementary clauses involving two identical predicates, one negated, the other not;

(b) other still more highly pruned variants of predicate resolution, applicable to sets of statements of particularly favorable form (e.g. to collections of 'Horn' disjunctions, i.e. those in which at most one one predicate term occurs with a positive sign in each disjunction of the collection, all other disjoined predicate terms occurring negated);

(c) resolution variants (e.g. *paramodulation*) which treat certain important operations (e.g. the equality operator) in special, particularly efficient ways;

(d) more specialized resolution-related schemes, e.g. algebraic-identity manipulation systems like that introduced by Knuth and Bendix, which exploit the special properties of statement sets consisting exclusively of equations.

Beyond these relatively general techniques, researchers have devised a growing assortment of decision algorithms for various branches of mathematics, e.g. the Tarski decision procedures discussed in Section 4, decision algorithms for purely additive integer arithmetic (Presburger), decision procedures for the purely Boolean theory of sets (Behmann), for the elementary unquantified theory of sets allowing the membership relator and the powerset operator, for various elementary parts of analysis, topology and geometry.

However, the general theorems described in Section 4 limit the utility of all these techniques by asserting that their computational cost must always rise prohibitively with modest enlargements in the general classes of statements with which they deal. For example, rule-of-thumb estimates concerning typical applications of the popular and very general resolution technique often indicate that even after pruning, and even if one starts with just ten or so initial statements, something like a 3-way branching in the possible pattern of operations can be expected to occur at each elementary inference step. It follows that discovery of a proof involving 14 successive elementary steps may involve search of as many as $3^{14}$ nodes of a tree of possibilities, a computation lying at the outer bounds of feasibility. Moreover, the branching ratio 3 appearing in this illustration can be expected to rise either if the proof to be developed starts with a somewhat larger set of initial statements (i.e. of 'axioms' or 'hypotheses'), or if structurally speaking this set of statements is exceptionally powerful (in the sense of allowing highly varied inference patterns, as for example in the case of the axioms of set theory.) It therefore seems very likely that fundamentally new ideas will have to be discovered before even the best known methods of this type become capable of producing proofs of as many as 20 elementary steps. All this is to say that even the best formal logic-manipulation techniques presently known still lack the human mathematician's uncanny ability to produce long and complex proofs by expanding a simple heuristic notion into a relatively undetailed, and probably not entirely accurate, proof sketch, which is then further expanded and amended into a full and accurate final proof.

Without such an ability, it will remain impossible to integrate the growing collection of known logic-manipulation techniques into a general tool capable of routine application to a broad variety of symbolic analysis or synthesis problems. Moreover, this basic limitation must also be read as a limitation on the power of all other known symbolic manipulation techniques that are general enough to be relevant to the very fundamental problem of constructing formal mathematical proofs.

### 5.3.3. Expert Systems

Many of the most active current attempts to commercialize ideas drawn from artificial intelligence research have focused on so-called 'expert systems'. Since systems of this kind are very much less general than deeper symbolic manipulation systems, such as predicate logic or graph-search systems, which aim at more significant levels of self-organization, there is a much better chance of bringing them to acceptable efficiency levels.

Expert systems typically concern themselves with small fixed sets of assertions relevant to a limited subject domain within which they aim to make simple but useful deductions. For example, the goal of a medical expert system might be to arrive at one of a finite number of possible conclusions drawn from a list such as 'Penicillin should be prescribed', 'Streptomycin should be prescribed' ,.., possibly supplemented by one or more explanatory diagnostic conclusions drawn from a list of possibilities such as 'the bacterial agent of the disease is pseudomonas', 'the bacterial agent of the disease is salmonella' ,... . The internal core of such a system, its so-called *inference engine,* ordinarily deals only with elementary statements of fixed form drawn from a finite list of possibilities. In our medical example, these might include 'inflammation is present', 'fever is present', 'the symptom site is lower abdomen', 'the white-cell count is elevated', and so forth. Typically, expert systems regard such assertions as unanalyzed logical atoms, subject only to elementary propositional manipulation, or perhaps some elementary form of probabilistic manipulation, rather than to any more penetrating predicate reasoning. Hence the 'expertise' which the system embodies is actually expressible by a collection of straightforward propositional or probabilistic rules in which the elementary assertions recognized by the system appear as indivisible units, e.g. 'If inflammation is present and the white cell count is elevated and the bacterial agent of the disease is salmonella then streptomycin should be prescribed'. In more sophisticated expert systems, which supplement inference rules of this bald propositional form by allowing probabilistic rules, the inference engine will associate some 'probability' or other numerical score, rather than a simple Boolean truth-value, with each of the elementary statements which it recognizes and with each of its inferences.

The assertions manipulated by such systems typically divide themselves into three subclasses:

(a) final conclusions, of interest to the end-user of the system, which are to be confirmed or rejected;

(b) elementary items of evidence, concerning which the system queries the user interactively; and

(c) intermediate assertions, which play an internal role in the inference engine's logical manipulations, but which can be externalized when the system is called upon to explain its remarks or deductions.

The system queries its users progressively concerning all relevant elementary evidence items (b), and employs the answers supplied to draw elementary Boolean (or somewhat more sophisticated probabilistic) conclusions concerning intermediate propositions (c) and final propositions (a). Type (a) propositions are what the user wants as system output and are presented to him in appropriate form and sequence.

The most rudimentary systems of this kind need not differ much from those questionnaires, familiar from popular magazines, which ask their readers to answer 'yes' or 'no' to a list of fairly obvious questions, each of which contributes a score of so-and-so-many points plus or minus to the outcome of some such query as 'Rate Yourself as a Parent'. However, a substantial level of function can be hung on these rudimentary frameworks:

(1) Expert systems can include attractive natural language and/or graphic interfaces.

(2) Instructions for carrying out any diagnostic procedures or tests required to answer queries of type (b) can be stored in such systems and made available when the system user is asked the corresponding questions. Specialized editors, databases, visual aids, and modeling systems relevant to a system's application domain can also be provided.

(3) Questions can be cleverly sequenced rather than simply being asked in fixed order. If evidence already supplied allows such a question to be answered either definitively or with high probability, or if it makes a question irrelevant to the type (a) final conclusions at which an expert system aims, the question can be suppressed.

(4) A system's user can be allowed to ask how particular final or intermediate conclusions were arrived at, in response to which the system can display its internal Boolean or probabilistic deduction steps, along with the built-in rules justifying these steps, in forms calculated to aid user comprehension.

(5) In some application areas, special deduction rules or other symbolic manipulations going beyond the merely propositional will be possible. For example, an expert system oriented toward chemical syntheses or analyses may be able to manipulate structural descriptions of molecules; an expert system dealing with electrocardiograms may be able to ingest raw cardiographic data and apply sophisticated spectral analysis or other pattern-matching procedures to it. The power of expert systems which include special techniques of this sort may rise substantially above the level attainable by primitive Boolean inference.

Overall, we can say that expert systems enhance their pragmatic applicability by narrowing the traditional goals of artificial intelligence research substantially, and by blurring the distinction between clever specialized programming and use of unifying principles of self-organization applicable across a wide variety of domains. This makes their significance for future development of deeper artificial intelligence technologies entirely debatable in spite of their hoped-for pragmatic utility.

### 5.3.4. Knowledge Representation

The phrase 'knowledge-based system' has become popular among scientists seeking to apply artificial intelligence research, and the associated dictum that 'finding appropriate representations of knowledge is one of the most basic problems of the artificial intelligence field' has often been propounded. Unfortunately, it is hard to identify any data structures created by the artificial intelligence research community that are other than superficial. Aside from clever internal implementations of such languages as LISP (which no one would consider 'knowledge representation' in any specific sense), no structures more advanced than simple pointer networks seem to have been proposed. Of course such networks are quite familiar from many other applications as ‘graphs’ or simply ‘mappings’. They involve nodes that are little different from the ‘records’ of standard data processing. This contrasts strongly with other branches of computer science, in which many quite ingenious data structures have been developed. In these fields, numerous successful examples have given the phrase ‘data structure design’ a mature technological meaning: any way of storing one or more abstract data entities in a manner which significantly accelerates the speed with which some well-defined battery of operations can be applied to these entities defines a significant data structure. Examples include B-trees, AVL-trees, Fibonacci heaps, compressed balanced trees, and many others. The underlying aim of artificial intelligence researchers in regard to ‘knowledge representation’ is of course the same as that of other computer scientists, namely to find data representations that can be used to accelerate the symbolic calculations that they would like to perform. However, progress toward this goal has stalled, since no acceptable formulation of the abstract structures to be implemented, or of the operations to be performed upon them, has yet become available. The one possible exception is use of ‘semantic nets’ for fast retrieval of items associated with other data items used as keys, a standard programming technique that artificial intelligence research actually has used in a manner no more sophisticated than is now common in database practice.

### 5.3.5. Learning

As stressed previously, one of the profoundest goals of artificial intelligence is to make computers capable of learning, i.e. capable of using disorganized information fragments to construct organized structures on which they can take action. Broad success with this one point would be almost equivalent to full realization of the subject’s aspirations. Unfortunately, almost nothing has yet been accomplished toward this bold goal. The disappointments encountered are typified by the variety of schemes that have been tried for allowing a computer to acquire the grammar of simple formal languages by exposure to sets of grammatical strings belonging to such languages. Although various faintly encouraging theorems have been proved concerning the asymptotic convergence of various learning algorithms to a desired grammar given sufficiently large numbers of positive and negative sentence examples, the enormous number of candidate grammars that present themselves have frustrated all practical use of this scheme. Related experiments include attempts to discover the simplest possible Boolean expression for a subset S of the set of all computer words of fixed length (whose bits can be thought of as representing true/false attributes of some class of objects or scenes). The input to such experiments are sets of positive and negative examples, or information concerning ‘near misses’ which can be given by stating the distance (measured in bits wrong) of each sample word from the nearest member of S. However, beyond various fragmentary heuristics, neither a practical approach to this problem nor any understanding of its inherent computational cost is available.

Other more trivial data-acquisition capabilities have been demonstrated and can be regarded as learning of a sort. For example, it is possible for a computer equipped with an image digitizer to acquire pictures of objects successively presented to it, then to calculate and store shape parameters for the boundaries of these objects, and subsequently to recognize the same objects when seen in other positions (at least, this is possible for favorable classes of objects). Perhaps this can be regarded as a rudimentary form of learning. Other techniques sometimes described as automatic learning involve use of data-derived statistics to adjust numerical parameters internal to a program. An even simpler possibility is to supply internal program constants progressively and interactively rather than all at program definition time. An example of this limited and artificial type of 'learning' would be a string analysis program, designed to be aware of the distinction of single characters (which it extracts internally from character data fed to it) into vowels and nonvowels, but not told initially which characters are which. Such a program can trivially emit an enquiry about each newly encountered character, following which the character can be inserted into one of two internally maintained sets, making subsequent enquiry unnecessary. The reader may or may not wish to regard this as true 'learning', since in much the same sense, one could view any menu-driven program which elicits and stores information concerning its user's preferences as a program which learns.

### 5.4. A Comment on Methodology

As might be expected of a young scientific discipline concerned with new, profound, and enormously attractive problems, the methodological level of research in artificial intelligence is often low. This contrasts with the situation in those other branches of computer science in which it has proved possible to define reasonably specific and feasible computational goals in a manner independent of the the techniques known at any given moment for trying to reach these goals. Where this has been possible, clear challenges have come before algorithm designers (who then often have found sophisticated and sometimes quite unexpected ways of computing important quantities) and computational complexity theorists (who seek to clarify the options open to the algorithm designer by proving theorems concerning the minimum computational cost of particular operations.) The systematic work flowing from this clarification of goals has substantially increased the maturity of other branches of computer science. Disappointingly, more primitive approaches have persisted in artificial intelligence research. Too many publications in this field simply describe the structure of some program believed by its authors to embody some function mimicking some aspect of intelligence, but aside from this having no definition other than the particular procedures of which it consists. It is often impossible to determine just what such a program really computes, or whether it does so with acceptable or catastrophic efficiency, or whether some other much more efficient technique might not have computed essentially the same thing. Still more primitive but nevertheless common publications consist of lightly or heavily edited traces of some program's internal activity, accompanied by author comments on felt similarities between this activity and the author's personal theory of mental function; a form of report which often leaves its reader without much understanding of what the program described is really doing, or how, or with what limitations. The unsatisfactory nature of all this is frequently compounded by the rudimentary syntax of the LISP notations in which such programs are commonly expressed, which readily confounds trivialities with profundities. Until these signs of immaturity disappear it will be hard to regard the field as embodying much mature technology.

## 6. Artificial Intelligence and the Development of Programming Languages

As emphasized above, the most fundamental goal of artificial intelligence research is discovery of principles facilitating the integration of initially fragmented material into useful organized structures. This is also a fundamental aim of the programming language designer, who seeks languages that make it easy to use small independent code fragments to define complex processes. Such languages eliminate troublesome sources of programming error and can increase programming speed very considerably. For this reason, and because artificial intelligence researchers have regularly grappled with unusually complex programming problems, their work has been a particularly fruitful source of advanced programming concepts.

A few of the most significant ideas of this kind are worth noting. The *LISP* language developed early in the history of artificial intelligence research introduced powerful means for defining entirely general and flexible data structures, and, since these also could be used to represent the particularly simple externals of the language, provided an environment in which other still more advanced programming languages could easily be implemented for experimental use.

*Rule-based* programming aims to eliminate programmer concern with operation sequencing by allowing operations to be executed whenever corresponding enabling conditions are met, for which purpose statements having approximately the form

*WHENEVER condition DO operation END*

are provided. *Backtracking* simplifies the execution of complex explorations by allowing exploration to be routed along multiple parallel branches. The simplest way of providing this semantic facility is through a *choice operation* having some such syntactic form as

$$x: = ONE\_OF\ s$$

where $s$ is a set. When executed by a process $p$, this operation can create as many independent copies of $p$ as the set $s$ has elements, and in each of these new processes a different element of set $s$ should be assigned as the value of the element $x$. Finally, if the set $s$ is empty when the *ONE_OF* operation is executed, the process $p$ should be terminated, leaving sibling processes created by prior *ONE_OF* operations to continue execution.

Various artificial intelligence languages more advanced than LISP have emphasized use of these three semantic operations, plus others, in various combinations. For example, in a language which provides both recursion and backtracking, iterations (*'DO'* statements) and explicit conditional statements (*'IF'* statements) are both superfluous features. Recursions can be used to re-express iterations, and conditionals can be expressed in terms of backtracking, by creating a separate process to execute each branch of the conditional, and then immediately terminating those processes which correspond to failed conditions. After elimination of all iterations and conditionals every program reduces to a sequence of definitions of recursive procedures, and each such procedure reduces to a linear sequence of simple assignments. It then becomes possible to regard any elementary assignment, (e.g. $x := y + z$ ) as an operation which tests some corresponding elementary relationship (i.e. $x = y + z$) and if necessary assigns a value satisfying this relationship to any variable or variables appearing in the relationship and not possessing any previously specified value. This has the advantage that a relationship like $x = y + z$ can trigger either the assignment $x := y + z$ (if $x$ has no prior value) or $y := x - z$ (if $x$ has a prior value but $y$ does not). If the call on a procedure $P$ (or more properly, successful return from a call on $P$) is viewed as a kind of logical 'conclusion' and the linear sequence of statements $A_1, \ldots, A_n$ constituting the body of $P$ is regarded as the set of 'hypotheses' of this conclusion, the definition of the procedure $P$ can be written in a notation such as

$$A_1 \& A_2 \& \cdots \& A_n \rightarrow P, \tag{1}$$

which gives programs consisting of such procedures the flavor (though not the full reality) of sets of statements in predicate logic. The fact that multiple redefinition of a procedure or procedures is harmless in a language providing backtracking (invocation of such a procedure can simply create multiple parallel processes, in each of which just one of the potentially relevant procedure definitions is invoked) reinforces the resulting resemblance to predicate logic, since it allows 'implications' of the form (1) to be inserted into a program freely and in arbitrary number. These semantic reflections underly the definition of the PROLOG language, which some artificial intelligence researchers have recently come to view as a significant addition to the older and better established LISP language.

Though programming in one of the advanced programming languages reviewed in the preceding paragraphs is sometimes described as ‘application of artificial intelligence technology’, it should be realized that these languages only facilitate manual expression of complex procedural and declarative structures, but do not embody any real principle of self-organization in and of themselves. Moreover, they all pay a price in efficiency for their generality: if used carelessly all the most advanced of these languages, including the rule-based and PROLOG-like systems, make it very easy to describe catastrophically inefficient computational processes. For this reason, the clean logical basis of these languages is often disrupted by inclusion of irregular efficiency-enhancing mechanisms of very different flavor, often making their effective use as full of pitfalls as ordinary programming languages of lower aspiration.

Since the fundamental goals of artificial intelligence research are far deeper than those of programming language design, extensive elucidation of its problems simply by design of some appropriate programming language is not to be expected.

### 6.1. Automatic Programming

The term ‘automatic programming’ refers both to fully computerized generation of programs from initial problem specifications expressed in entirely abstract, logic-like terms, and to automated improvement of program efficiency. Efficiency improvement can be realized by automatic transformation of less efficient into more efficient algorithms, or by automatic generation of detailed program versions in efficiency-oriented programming languages (such as Pascal or Ada), starting from considerably more concise ‘specifications’ written in a programming language such as (PROLOG or SETL) having much higher semantic level.

Proofs in some (‘intuitionistic’) logical formalisms can be compiled automatically into (highly inefficient) programs. Ordinarily, however, the problem of generating programs from problem statements written in a formalism close to that of logic is very similar to the problem of generating proofs in logic automatically, and hence is subject to the pessimistic assessment offered at the end of the preceding subsection.

Automatic improvement of program efficiency is a related problem which has attracted considerable attention, much of which has concentrated on the possibility of exploiting libraries of optimization tricks of the kinds most commonly used by human programmers. One typical device of this kind is use of *formal differentiation*. In this technique, one keeps up-to-date values of expressions, used within program iterations, that would otherwise have to be recalculated repeatedly at substantial computational cost; the expression values required are then kept current by updating them, hopefully at substantially lower expense, whenever any of their arguments changes.

This is one of the most promising techniques for automatic program optimization at a very abstract level, and can readily be seen to account for important aspects of the approach to manual development of efficient programs actually employed by programmers in many cases. However, systematic work on this method (by Paige and others) during the last years has shown that effective application even of this particularly favorable approach raises too many deep problems for its automatic application by any known method to be feasible. The difficulty is that even for programs which visibly fit the 'formal differentiation' stereotype, efficiency improvement generally depends on knowledge of secondary logical constraints concerning possible program states at specified program points. These constraints are typically deep enough to defy automatic verification, and also complex enough for their full statement to discourage programmer involvement. Here again we have a situation in which the computer's inability to deal efficiently even with intuitively simple sets of logical statements raises a significant obstacle to progress. Similar objections apply to other proposed techniques for automatic program improvement, many of which raise much the same problems of exploding combinatorial search of symbolic structures as are involved in automatic discovery of mathematical proofs, often in particularly virulent form, because both the program texts which must be processed and the vocabulary of transformations applicable to such texts are considerably larger than the small examples ordinarily considered in the research literature on automatic discovery of proofs.

A consequènce of all this is that only relatively rudimentary transformations have found profitable application to automatic improvement of program efficiency. Normally such automatic optimization only pays for itself when a small number of relatively superficial techniques can be applied inexpensively to extensive computer texts, so as to eliminate wholesale inefficiencies introduced by prior steps of automatic processing, e.g. by straightforward compilation or macro-expansion of source text. Program optimization of this practical form has more the flavor of large-scale symbolic data processing than with artificial intelligence research (though partial affinity with some of the deeper goals of artificial intelligence research can be discerned). Even the intermediate-level problem of automatically introducing data structures into program texts written in very high level languages, so as to raise program efficiency to levels that human programmers can routinely reach, lies somewhat beyond our present grasp.

## 7. Moral Limits

Successful construction of artificial intelligences would affect the human environment profoundly. If artificial intelligences can be created at all, there is little reason to believe that initial successes could not lead swiftly to the construction of artificial superintelligences able to explore significant mathematical, scientific, or engineering alternatives at a rate far exceeding human ability, or to generate plans and take action on them with equally overwhelming speed. Since man's near-monopoly of all higher forms of intelligence has been one of the most basic facts of human existence throughout the past history of this planet, such developments would clearly create a new economics, new sociology, and a new history.

Part of the opposition which certain humanist thinkers have made to the entire notion of artificial intelligence stems from this fact. They express the amorphous unease of a much broader public. The fear is that the whole fabric of human society, which at times seems terrifyingly fragile, may be torn apart by enormously rapid technological changes set in train by artificial intelligence research as it begins to yield its major fruits. It is for example possible to imagine that would-be dictators, small centrally placed oligarchies, or predatory nations could exploit this technology to establish a power over society resting on robot armies and police forces independent of extensive human participation and entirely indifferent to all traditional human or humane considerations. Even setting this nightmare aside, one can fear various more subtle deleterious impacts, for example rapid collapse of human society into a self-destructive pure hedonism once all pressures, and perhaps even reasons or opportunities, for work and striving are undermined by the presence of unchallengeably omnicompetent mechanisms. Certainly man's sense of his own uniqueness is bound to be impaired, and he may come to seem in his own eyes little more than a primitive animal, at best capable of some fleeting enjoyments.

Successful response to such developments when and if they begin to accelerate will require humanity to reaffirm its spiritual solidarity and to close ranks across class, ethnic, and national boundaries. Conservative prudence needs to be combined with graceful and constructive adaptation to deep and rapid change. Once man is generally seen as an intelligent mechanism, and mechanisms as intelligent as man regularly flow forth from factories, what limits must be set to the manipulation either of man or his created mechanisms? What regulations and social assumptions will prove appropriate to a world in which work, except as hobby, has come to an end? These questions, which even science fiction has as yet explored only occasionally, are likely to rush upon statesmen, philosophers, and theologians within just a few centuries.

## 8. Bibliography

D. Ballard and C. Brown, *Computer Vision*, Prentice-Hall Publishers, 1982.

A. Barr and E. Feigenbaum (editors), *The Handbook of Artificial Intelligence* (3 volumes), Heuristech Press, Stanford, 1982.

M. Boden, *Artificial Intelligence and Natural Man*, Basic Books, 1977.

M. Brady et al. (editors), *Robot Motion: Planning and Control*, MIT Press, 1982.

C.L. Chang and R.C. Lee, *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, 1973.

M. Davis and E. Weyuber, *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*, Academic Press, 1983.

H.L. Dreyfus, *What Computers Can't Do*, Harper and Row Publishers, 1972.

E. Feigenbaum and J. Feldman (editors), *Computers and Thought*, Krieger Publishing Co., Malabar, Florida, 1981.

J. Feldman, *Memory and Change in Connection Networks*, Rochester University Computer Science Technical Report 96, December 1981.

J. Ferrante and C. Rackoff, *The Computational Complexity of Logical Theories*, Springer Publishers, 1979.

F. Hayes-Roth, D. Waterman, and D. Lenat, *Building Expert Systems*, Addison-Wesley Publishers, 1983.

D.O. Hebb, *The Organization of Behavior*, John Wiley Publishers, 1949.

D.H. Hubel and T. Wiesel, *Brain Mechanisms of Vision*, Scientific American, September 1979, pp. 150-162.

M. Jacobsen, *Developmental Neurobiology*, Plenum Publishers, 1979.

E. Kandel, *The Cellular Basis of Behavior*, Freeman Publishers, 1976.

S. Kuffler, J. Nicholls, and A. Martin, *From Neuron to Brain*, Sinauer Publishers, Sunderland, Maine, 1984.

N. Sager, *Natural Language Information Processing: a Computer Grammar of English and its Applications*, Addison-Wesley Publishers, 1981.

R. Schank and C. Riesbeck (editors), *Inside Computer Understanding*, L. Erlbaum Publishers, 1981.

J. Siekmann and G. Wrightson, *Automation of Reasoning*, Springer Publishers, 1983.

C.Y. Suen and R. De Mori, *Computer Analysis and Perception*, Volume 3: *Auditory Signals*, Chemical Rubber Co. Press, 1982.

J. Weizenbaum, *Computer Power and Human Reason: From Judgement to Calculation*, Freeman Publishers, 1976.

P. Winston, *Artificial Intelligence*, Addison-Wesley Publishers, 1984.

P. Winston and R. Brown (editors), *Artificial Intelligence: an MIT Perspective* (2 volumes), MIT Press, 1979.

This book may be kept

FOURTEEN DAYS

A fine will be charged for each day the book is kept overtime.

NYU COMPSCI TR-212
Schwartz, Jacob T
The limits of artificial
intelligence. c.1

| | |
|---|---|
| FEB 1 0 1992 | 2. Lil |
| | |
| | |